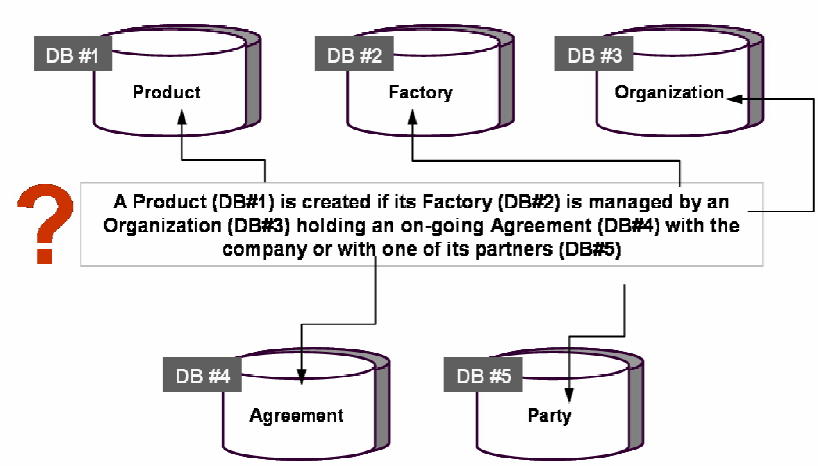
Data Flow and messages management

How to give to Business Users an autonomy in their daily data flow and messages management? How to streamline your integration platform by avoiding errors within data flows and messages?

Are links between your data killing your data flow and messages validation and traceabiliity?



This architecture brings a dangerous lack of data flow and messages integrity, data lineage and audit trail since all business logics are hard-coded, including referential integrity constraints spanning databases. This situation provides a lack of alignment with business regulations and IS governance best practices requiring full data flow and messages traceability such as CobiT, Sarbanes Oxley, Solvency II, IAS-IFRS, etc. and applied within integration platform with standards such as SWIFT, Fpml, Acord, EDI, etc.

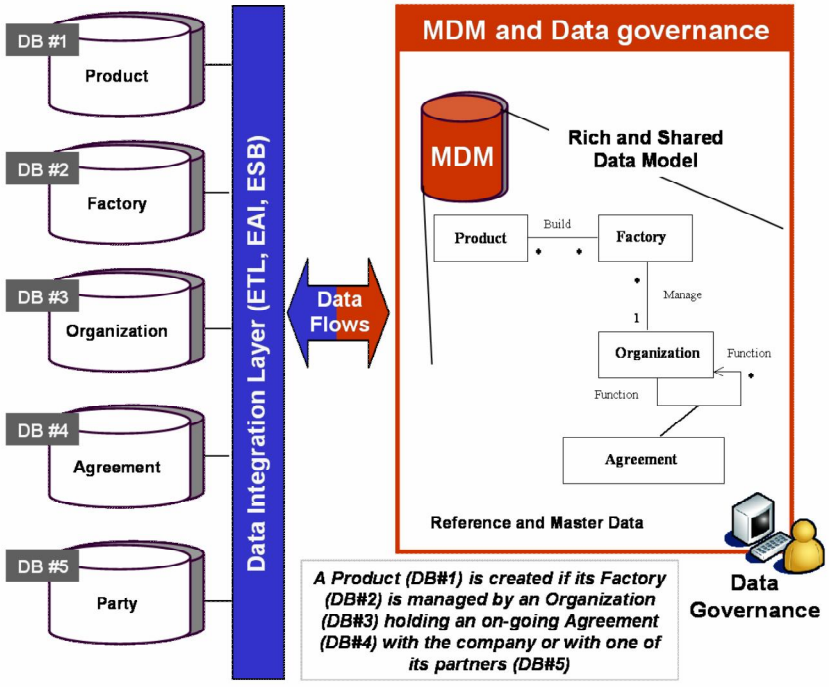
Most of the time, a company and its IT department face an acute concern when exchanging messages between systems within its boundaries and outside. The question raised is:
Where are located business rules enforcing integrity constraints applied to make safe associations between Business Objects embedded within data flows, it means within messages sent to consuming systems?

This question is very complicated when data is scattered across several physical databases. In left figure, every business object is located in a dedicated physical database which prevents from managing integrity constraints at the level database engines. Sadly, in a real system, data duplications would also be present, which would be an additional difficulty to tackle. Indeed, in this situation, data integrity mechanisms provided by every database system are no longer usable and then checking and overseeing messages become a real nightmare.

Most of the time, business rules required to manage integrity between business objects are hard-coded within the data integration layer, such as an ESB. These rules are handled by IT specialists only. They are also duplicated and dispersed within every functional silo, inside every database with triggers and other hard-coded software approaches.

To tackle this concern, MDM and Data governance come into play and complement with the integration platform to enforce messages validation and traceability, with a solution open to business users.

MDM and Data Governance are established to enforce data flow and messages validation and traceability



First of all, a shared Data Model must be established. This Common Information Model (CIM) is a Semantic Data Model spanning physical databases boundaries. It describes all links between business objects embedded with data flow and messages whatever their locations within silos in a company and outside. This model is obtained through an iterative lifecycle design, without any tunnel effect or big-bang approach. To get more information about modeling procedures applied to achieve this goal, please visit our sister community MDM Alliance Group. **When the integration platform relies on standard business languages such as Fpml, Swift, Acord, etc. the necessary Common Information Model is quickly available as a company uses the UML data models and its derivation into XML Schema of these standards directly. To really achieve this approach, it is important to note that a Model-driven MDM must be used. It means a MDM able to absorb any domains of data models, whatever its complexity and business targets: this is really far away from PIM (Product Information Management), CDI (Customer Data Integration) or other MDM-like solutions based on a frozen DDL (Data Description Language).**
Then, the CIM is fully implemented within a Master Data Management (MDM) to save information required to manage and oversee data flow and messages: **business objects identifiers (Primary and Foreign Keys), providers and consumers identifiers, referential integrity constraints (outcomes of business rules), technical headers and other data body depending on requirements**.

Data Flow and messages management

How to give to Business Users an autonomy in their daily data flow and messages management? How to streamline your integration platform by avoiding errors within data flows and messages?

The MDM must provide Business Users with Data Governance features such as querying, version and permissions management, authoring when it is needed to complement data flow and messages, full business audit trail to guarantee data flow traceability and data lineage, etc.

Could a MDM within the integration platform bring troubles of performance?

To tackle this point, three types of integration are used and can live together to fully manage performance issues:

First - Synchronous integration
Data flow and messages are checked and saved within the MDM repository before sending to consumers.

Second – Partially synchronous integration
Data flow and messages are checked only and then pushed to consumers. In other word, data flow and messages storage within the MDM repository is done after sending to consumers

Third – Asynchronous integration
Data flow and messages are checked and stored after sending to consumers only.

How to choose within these three types of integration?

It depends on your data governance objectives. You can start with the easiest approach “Asynchronous integration”. There are no impacts on your current ESB architecture. Even with this integration, you obtain a full business audit trail guaranteeing a complete data flow and messages traceability, data lineage and errors reports. When data flow errors are detected you set up IS governance processes to re-establish a stable situation by benefiting from the MDM full business data flow and messages traceability. When needed you can involve your business users since you hold a business governance of data flow and messages, not only an IT audit trail not readable by business users.